

# Django Form 폼나게 쓰기

Django Form 원리부터 활용까지

강승형

sean.kkang@gmail.com

**Back to the Basic**

# 파이썬을 사랑하는 아재 개발자

발표자 소개

**Back to the Basic**



**STARBUCKS®**

안녕하세요. 스타벅스입니다.



Rewards



Gold Level

그료음료 쿠폰 획득 별까지  
2개 남았습니다.

**파이썬짱**

Inbox

e-Coupon

**Back to the Basic**

# 발표 소개

오늘의 이야기

**Back to the Basic**

# Django Form?

## Internal

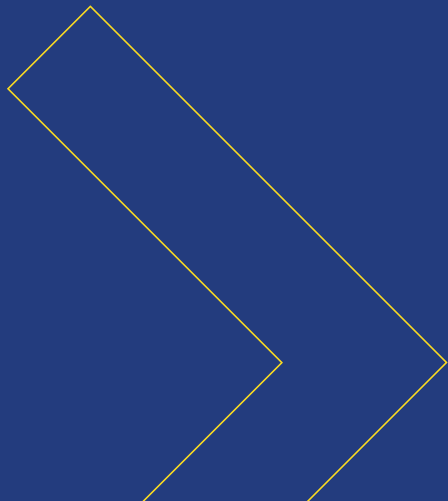
## ModelForm

## FormView, GenericView

**Back to the Basic**



# Django Form?



**Back to the Basic**

# Web의 Form 처리 추상화

**Back to the Basic**

**Name\*****Desc**

Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form의 동작원리를 살펴보고, Django에서 권장하는 Diangonic한 방법의 Form 사용법도 살펴보고, 실제 로직 구현시 부딪히게된 문제와 해결 방법을 공유할 예정입니다.

**Language\*****Back to the Basic**



# 성공 : Redirect

**Back to the Basic**

# 실패 : Error

**Back to the Basic**

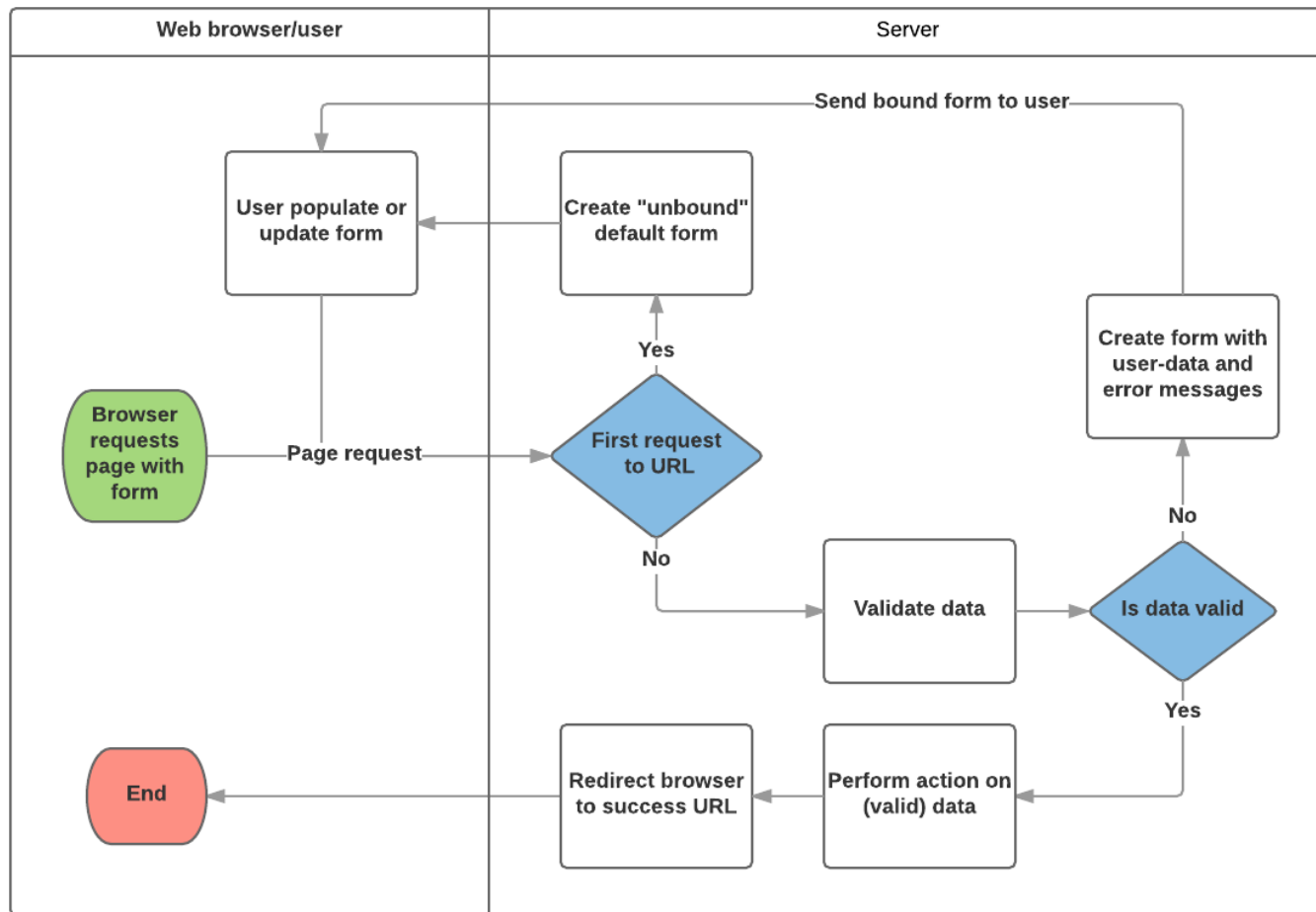
**Name\*****필수 항목입니다.****Desc**

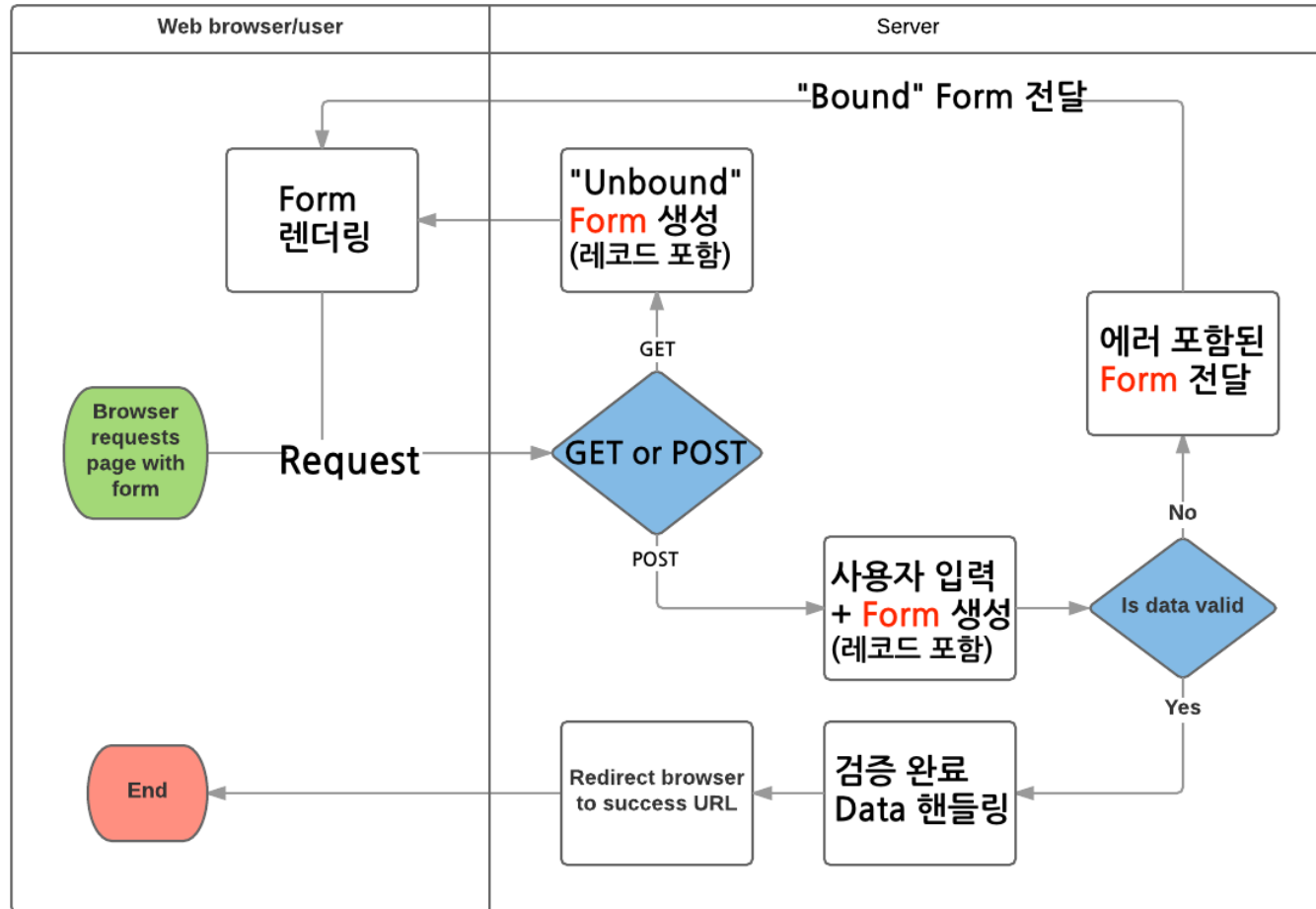
Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form의 동작원리를 살펴보고, Django에서 권장하는 Djangonic한 방법의 Form 사용법도 살펴보고, 실제 로직 구현시 부담히게된 문제와 해결 방법을 공유할 예정입니다.

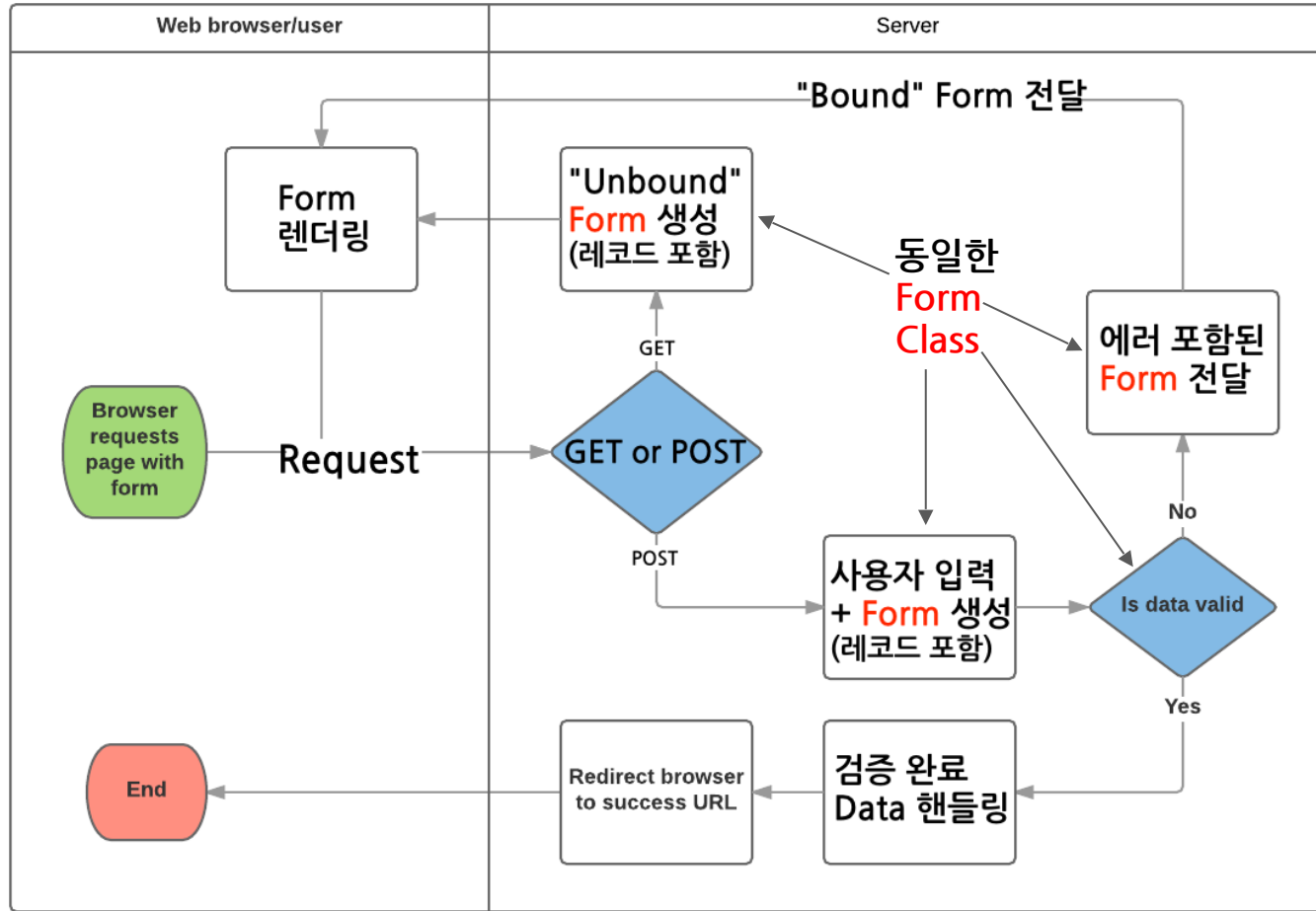
**Language\*****전송****Back to the Basic**

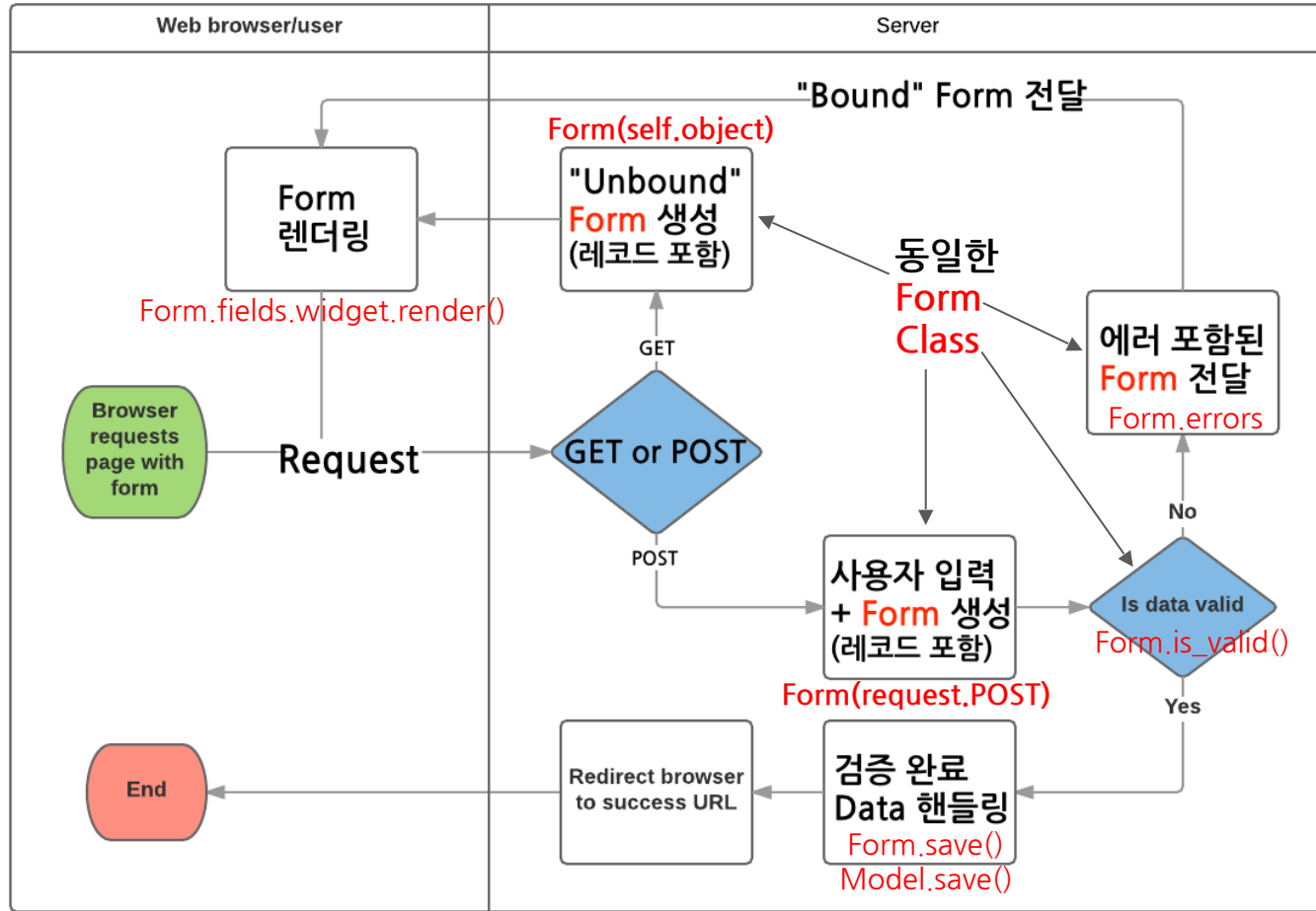
# Django의 Form 처리

**Back to the Basic**











# Django Form Class Inside

**Back to the Basic**

# Form의 pseudo-code

**class** Form:

fields = []

data = {}

initial = {}

cleaned\_data, errors = {}, []

**def** \_\_init\_\_(self, data=None):  
 self.is\_bound = data is not None

**def** is\_valid(self):  
 self.clean()  
 **return** self.is\_bound and not self.errors

**def** clean(self):

**for** name, field **in** self.fields:

**try**:

value = field.clean()

**except** ValidationError **as** error:

self.errors.append(error)

**else**:

self.cleaned\_data[name] = value

# Form의 pseudo-code

```
class Form
```

```
    fields =
```

```
    data = {
```

```
    initial =
```

```
    cleaned
```

```
    def __init__
```

```
        self.is
```

```
    def is_val
```

```
        self.c
```

```
        return
```



```
        fields:
```

```
        ()
```

```
        or as error:
```

```
        (error)
```

```
        [name] = value
```

# Field의 pseudo-code

**class Field:**

**widget** = **TextInput**

**validators** = []

**def** **\_\_init\_\_**(self, validators):  
    self.validators = validators

**def** **clean**(self, value):  
    self.run\_validators(value)  
    **return** value

**def** **run\_validators**(self, value):

    errors = []

**for** validator **in** self.validators:

**try**:

            validator(value)

**except** **ValidationError** **as** error:

            errors.append(error)

**if** errors:

**raise** **ValidationError**(errors)

# Field의 유효성 검증

```
class Field
```

```
    widget
```

```
    validate
```

```
    def __in
```

```
        self.v
```

```
    def clea
```

```
        self.r
```

```
        retur
```



```
        value):
```

```
        idators:
```

```
        or as error:
```

```
        or)
```

```
        r(errors)
```

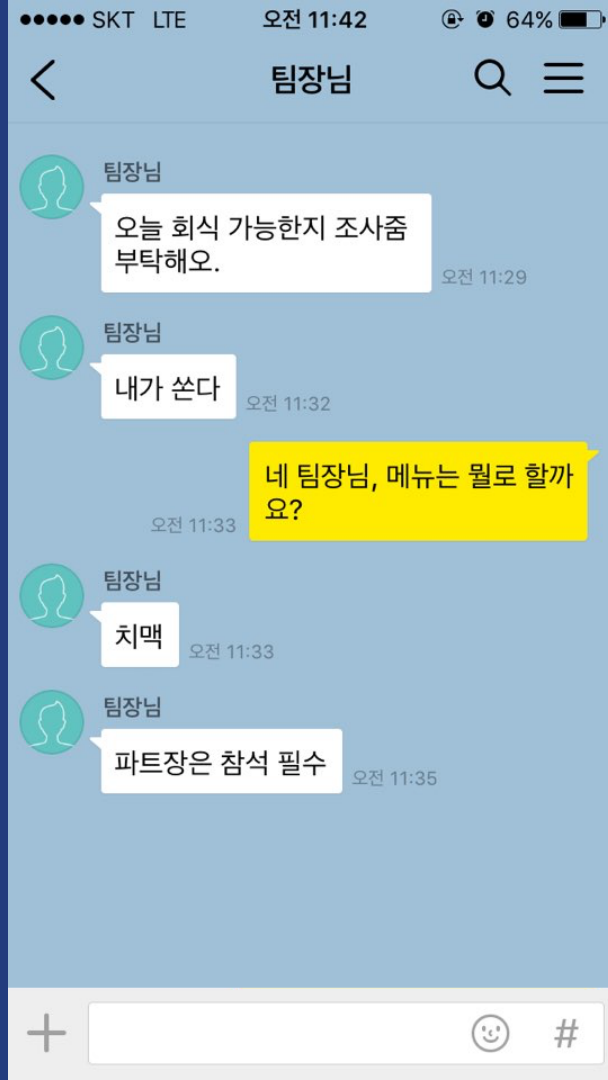
# Form The 개발팀 회식 이야기

개발팀 회식 = Form, 개발자 = Field

**Back to the Basic**

개발팀 회식 = Form

파트장 필수 =  
`Form.clean_파트장()`



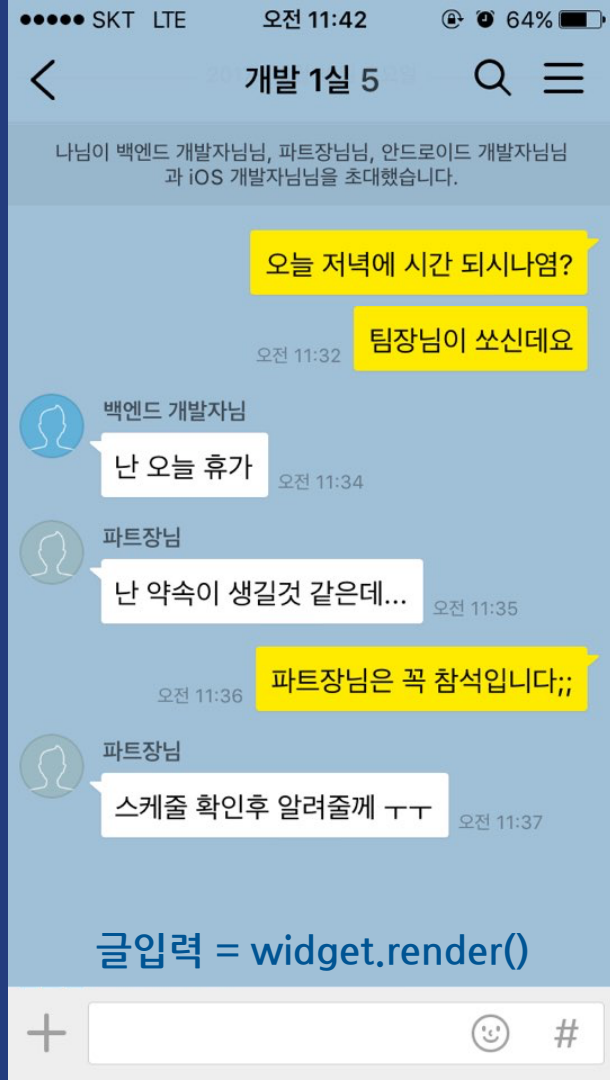
내가 쏜다 = Submit  
Data = 오늘, 치맥

**Back to the Basic**

fileds = [파트장,  
백엔드, 안드로이드, iOS]

raise Exception(['백엔드  
개발자님이 휴가래요', ])

백엔드\_Field(휴가=월)



회식가능? =  
Form.is\_valid()

파트장 필수 =  
Form.clean\_파트장()

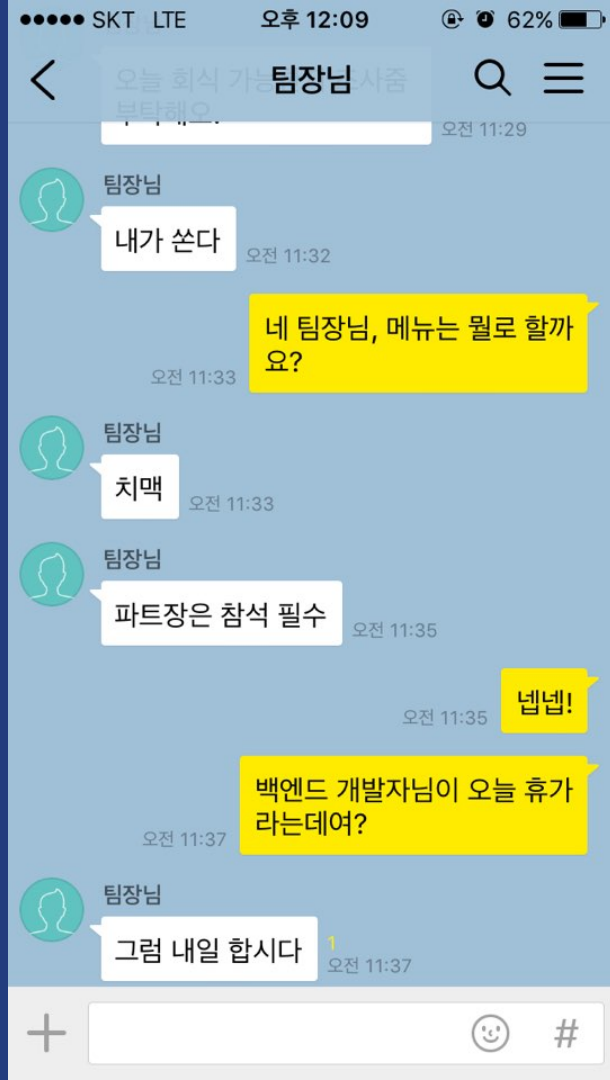
**Back to the Basic**



개발팀 = Form

팀원들 = Fields

Data = 내일, 치맥  
합시다 = Submit



Display Errors =  
`Form.form_invalid()`

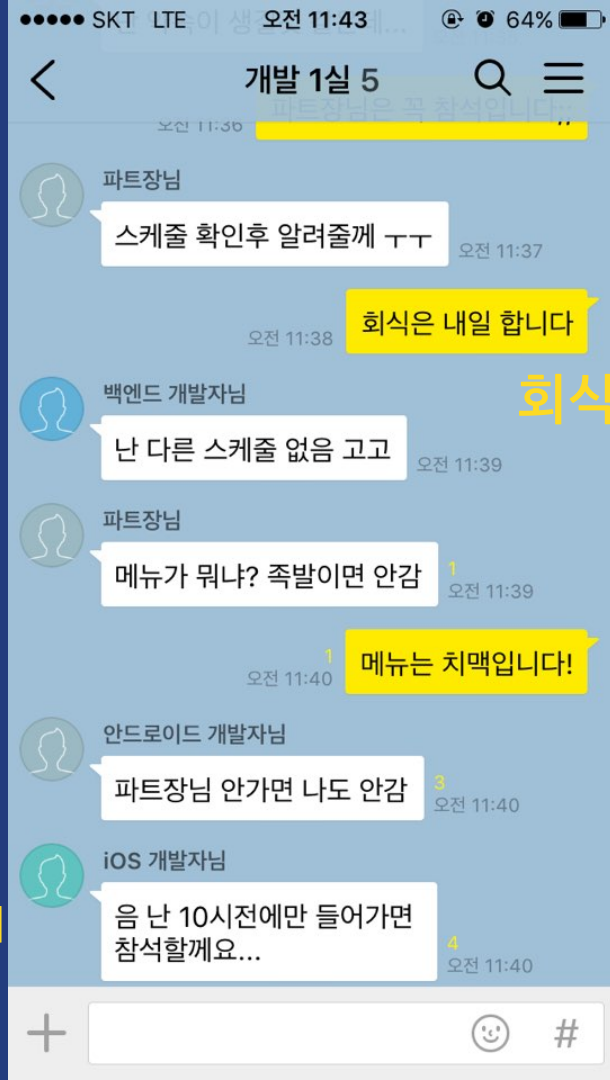
**Back to the Basic**

팀원들 = Fields

파트장.validators =  
[lambda 메뉴: !족발, ]

안드로이드.validators =  
[lambda 참석자: 파트장 in 참석자, ]

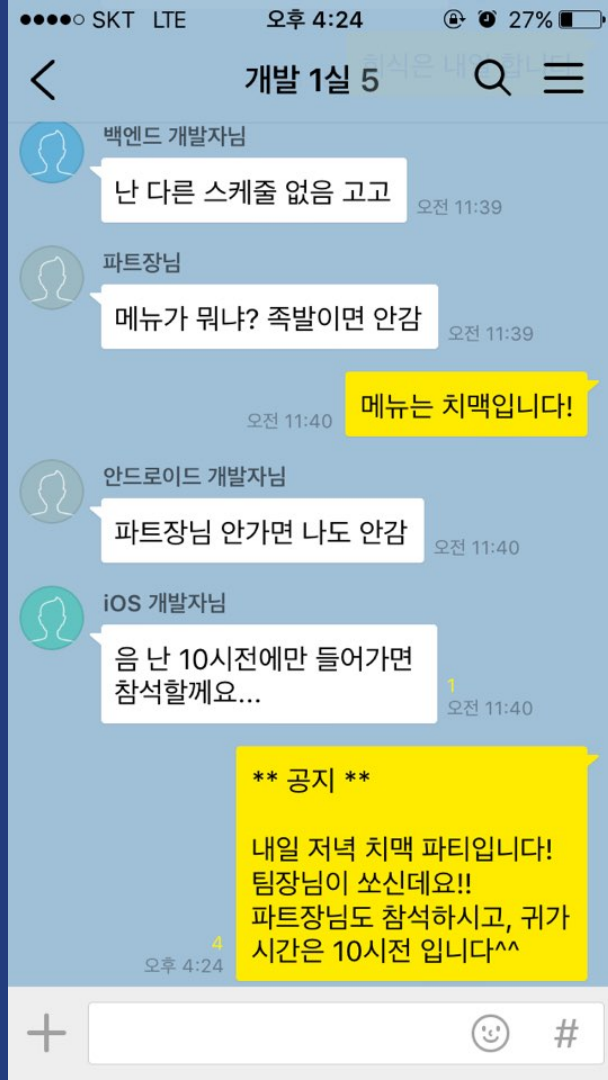
iOS.validators =  
[lambda 귀가시간: 귀가시간 < 22시, ]



회식가능? = Form.is\_valid()

파트장.clean() = 통과

**Back to the Basic**



Success  
= Form.form\_valid()

**Back to the Basic**

# Form의 일반적 사용

```
class MyForm(forms.Form):  
    field1 = forms.IntegerField(min_value=0, max_value=100000, required=True)  
    field2 = forms.CharField(max_length=10, validators=[regex], required=False)  
  
    def clean_field1(self):  
        cleaned_field1 = self.cleaned_data['field1']  
        if not self.additional_validate(cleaned_field1):  
            raise forms.ValidationError(  
                self.error_messages['field1_incorrect'], code='field1_incorrect',  
            )  
        return cleaned_field1
```

# Model Form?

**Back to the Basic**

# ModelForm = Form from Model

**Back to the Basic**

Form  $\approx$  Model

**Back to the Basic**

**Form.field  $\approx$  Model.field**

**Back to the Basic**



**Form.save()  $\approx$  Model.save()**

**Back to the Basic**

# Django Form 폼나게 쓰기

제목\*

Django Form 폼나게 쓰기

슬라이드 URL

동영상 URL

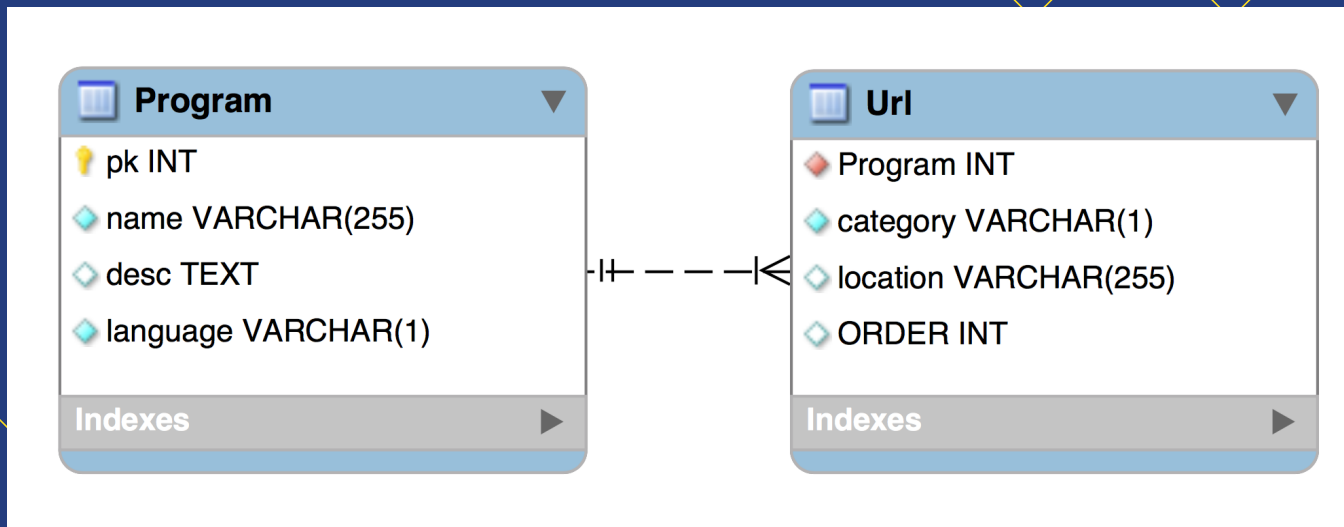
☒ 촬영, 녹화가 가능합니다.

설명



Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form의 동작원리를 살펴보고, Django에서 권장하는 Djangonic한 방법의 Form 사용법도 살펴보고, 실제 로직 구현시 부딪히게된 문제와 해결 방법을 공유할 예정입니다.

# 간단하게 Modeling



# 간단하게 Modeling

```
class Program(models.Model):  
    name = models.CharField(max_length=255, db_index=True)  
    desc = models.TextField(null=True, blank=True)  
    language = models.CharField(max_length=1, choices=(  
        ('E', _('English')), ('K', _('Korean')), ), default='E')
```

```
class Url(models.Model):  
    program = models.ForeignKey(Program)  
    category = models.CharField(max_length=1, choices=(  
        ('S', _('Slide')), ('P', _('PDF')), ('V', _('Video')), ) )  
    location = models.CharField(max_length=255)  
    ORDER = models.IntegerField(default=0)
```

# ModelForm은 초간단

```
class ProgramForm(ModelForm):  
    class Meta:  
        model = Program  
        fields = ['name', 'desc', 'language']
```

```
class UrlForm(ModelForm):  
    class Meta:  
        model = Url  
        fields = '__all__'
```

# Template도 초간단

```
<form method="post">  
    {% csrf_token %}  
    {{ form.as_p }}  
    <input type="submit" value="전송"/>  
</form>
```

Name:

Django Form 폼나게 쓰기

Desc:

Language:

**Back to the Basic**

# View는 복잡

```
class ProgramCreate(TemplateView):
```

```
    form_class = ProgramForm
```

```
    template_name = 'program_form.html'
```

```
    success_url = reverse_lazy('program:list')
```

```
    def post(self, request, *args, **kwargs):
```

```
        form = self.form_class(data=request.POST, instance=self.object)
```

```
        if form.is_valid():
```

```
            form.save()
```

```
            return HttpResponseRedirect(self.success_url)
```

```
        else:
```

```
            return self.render_to_response(self.get_context_data(form=form))
```

```
    def get_context_data(self, **kwargs):
```

```
        if 'object' not in kwargs:
```

```
            kwargs['object'] = self.object
```

```
        if 'form' not in kwargs:
```

```
            kwargs['form'] = self.form_class(instance=self.object)
```

```
        return super().get_context_data(**kwargs)
```



```
class ProgramCreateView(
    form_class = ProgramCreateForm,
    template_name = 'program/create.html',
    success_url = reverse_lazy('program:detail')
```

```
def post(self, request):
    form = self.form_class(request.POST)
    if form.is_valid():
        form.save()
        return HttpResponseRedirect(
        else:
            return self.render_to_response(
```



```
    context = {'instance': self.object,
               **kwargs})
```

# FormView로 간단히

```
class ProgramFormViewCreate(ModelFormMixin, FormView):  
    form_class = ProgramForm  
    object = None  
    template_name = 'program_form_formview.html'  
    success_url = reverse_lazy('program_formview:list')
```

# GenericView로 초간단

```
from django.views.generic import CreateView  
class ProgramGenericCreateView(CreateView):  
    model = Program  
    fields = ['name', 'desc', 'language']  
    success_url = reverse_lazy('program_generic:list')
```

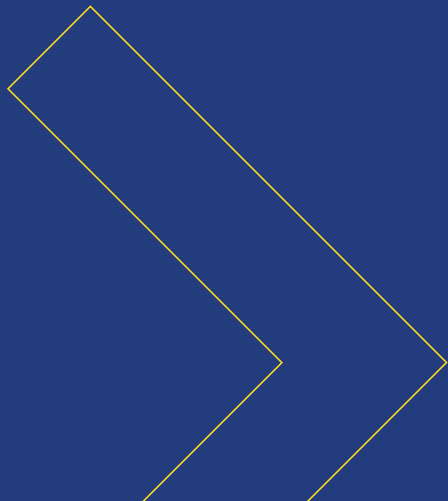
세상에! Form이 없어도 된다니!

# Formsets

**Back to the Basic**



# Formsets = a Set of Form



**Back to the Basic**

## Program

Name:

Django Form 폼나게 쓰기!

Desc:

Language:

## URL

Category:  Location:  순서:  삭제: ☐

Category:  Location:  순서:  삭제: ☐

Category:  Location:  순서:  삭제: ☐

# Inlinformset 사용하기

```
class ProgramFormSetViewCreate(ModelFormMixin, FormView):  
    form_class = ProgramForm  
    formset_class = inlineformset_factory(parent_model=Program, model=Url,  
                                          form=UrlForm, extra=1, can_order=True, can_delete=True)
```

```
def get_context_data(self, **kwargs):  
    context = super().get_context_data(**kwargs)  
    if 'formset' not in kwargs:  
        context['formset'] = self.get_formset  
    return context
```

# Inlinformset 사용하기

```
<form method="post">
```

```
{% csrf_token %}
```

```
{{ form.as_p }}
```

```
<h4>URL</h4>
```

```
{{ formset.management_form }}
```

```
{% for each_form in formset %}
```

```
    <p> {{ each_form }} </p>
```

```
{% endfor %}
```

```
<input type="submit" value="전송"/>
```

```
</form>
```



# django-crispy-forms

**Back to the Basic**

Name: Django Form 품나게 쓰기!

Django를 Full stack으로 사용하면서 가장 많이 고민  
Form에 대한 부분의 경험을 공유하기 위한 발표입니다  
본적으로 Django 내부의 Form의 동작원리를 살펴보고  
Django에서 권장하는 Djangonic한 방법의 Form 사  
도 살펴보고, 실제 로직 구현시 부담히게된 문제와 해결  
법을 공유할 예정입니다.

Before

Desc:

Language: 영어

전송

Name\*

Django Form 품나게 쓰기!

Desc

Django를 Full stack으로 사용하면서 가장 많이 고민해던 Form에 대한 부  
분의 경험을 공유하기 위한 발표입니다. 기본적으로 Django 내부의 Form  
의 동작원리를 살펴보고, Django에서 권장하는 Djangonic한 방법의  
Form 사용법도 살펴보고, 실제 로직 구현시 부담히게된 문제와 해결 방법을  
공유할 예정입니다.

After

Language\*

영어

전송

감사합니다.

<https://github.com/grepsean/pyconkr-2017-django-form>

**Back to the Basic**